Software quality management : ISO 9000, but not only

Balla, K.

Published: 01/01/1995

Document Version
Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

• A submitted manuscript is the author's version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

Citation for published version (APA):

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal?

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Download date: 02. Jan. 2019
Software Quality Management: ISO 9000, but not only

door
K. Balla

Report EUT/BDK/68
ISBN 90-386-0055-0
ISSN 0929-8479
Eindhoven 1995
Balla, K.

Software quality management: ISO 9000, but not only / door K. Balla. - Eindhoven: Eindhoven University of Technology, Graduate School of Industrial Engineering and Management Science. - (Report EUT/BDK, ISSN 0929-8479 ; 68)

With ref.
ISBN 90-386-0055-0
NUGI 689

Subject headings: software quality.
Software quality management: ISO 9000, but not only

Research report
Katalin Balla
Technical University Budapest
Department for Information Theory and Computer Science
e-mail: balla@inf.bme.hu

Abstract:
Following a brief presentation of the software quality oriented work and research done in Hungary, quality-oriented concepts, theories, Dutch results are presented, as learned during a 6 weeks stay at Eindhoven Technical University. Possibilities to apply these in Hungarian software industry are suggested. The paper concentrates on some basic concepts, ideas which suggest a new approach to the problem of software quality management. The Hungarian software industry being for the moment concentrated on quality management as suggested by ISO 9000, possibilities are shown to a differentiate application of this standard series. In the last chapter some testing techniques and tools, software metrics concepts and risk management concepts are presented.

1. Background

1.1. General Framework

Quality management is not a distinct subject among the Hungarian Technical University, Department for Electrical Engineering and Computer Science courses. Software quality management is even less. Some ideas are tried to be sketched, mainly as part of software engineering courses. These lectures concentrate on the concepts and use of structured system development methodologies, some project management methodologies and their computer-aided tools.

However, in Hungary a new type of quality- and software-quality-oriented knowledge starts to be needed. This “new type” can be characterised by saying, it's a “management-science-oriented” knowledge. I see the main reason of this need in the world-wide market competition, due to which many companies try to be registered according to ISO 9000. (Adopting ISO and EN Standards is also due to the trial and hope in EC integration.) Software industry in Hungary is dealing with this challenge. Due probably to the growing dimensions of the problems to be solved and to the deeper implication of information systems in more and more social fields, in software companies some internal desire for a better structured, more organised work in analysing, designing, programming, maintaining, documenting, testing software products and giving assistance for self-developed or adapted applications also exists. Although some national and many international quality organisations are present in Hungary, which, besides assessing, offer support for developing and introducing quality management systems (further: QMS), most companies prefer to develop their QMS by their own. The reason of this choice is in most cases the fact that assessing

1 Statements made in this paragraph reflect the author’s personal point of view.
2 Although “management-science” in Hungary is hardly considered to be a “science”.

itself is very expensive for Hungarian companies, not to speak about the price of assistance given. In the field of software these activities are provided for even a higher price, because software-quality-specialists have to be brought from abroad. In this situation, the management from one of the main representatives of software industry in Hungary, IQSOFT company, decided in summer 1993 to offer a scholarship for a Ph.D. student from the Technical University Budapest in order to build up, document and introduce an own software QMS.

The basic reason of IQSOFT’s management decision was - at that time - the emerging request to be ISO-certified, formulated by a (potential) foreign customer. However, it’s worth to note that IQSOFT has been established in 1990 by people working at the National Computer Science Research Centre, many of whom have been implicated in the former software-quality-oriented research. (Regarding the presence of actual IQSOFT employees in the overall former software-quality-oriented research see Annex 2.) One can understand that besides of joining the nowadays very popular (and market-requested) trend of ISO certification, a part of the IQSOFT employees has been formerly familiarised with the concept and importance of software quality.

The existing knowledge reflected the former general Hungarian concept: approaching software quality through the product-characteristics. Some negative experiences have been added to this knowledge: results of the formerly done research have not entered the “every-day life” of software developers, trends for test-automation, validation seemed to fail. The formerly existing process-oriented approach (e.g. using prescribed steps and documents in programming) also seemed to give no results.

These, I think, were the reasons why in September 1993, when I started my work at IQSOFT, most colleagues were rather sceptic regarding the usefulness and real help provided by a software QMS. However, everyone seemed to agree to the management’s decision to obtain ISO certification, but, at least for the beginning, the majority of colleagues believed that software quality management is a “needed hurtful thing” to be done in order to keep the company in the market.

At IQSOFT no formal quality management department has been established, and the “quality manager” herself was not an employee of the company. However, establishing a distinct department for quality management is being discussed currently. This shows the understanding of the importance of this kind of activity and understanding that is cannot be done “part time”, by an “outsider”.

1.2. THE PRACTICAL APPROACH

My first approach to the problem was the trial to understand ISO 9000 (especially 9001 and 9000-3) terminology and prescriptions. Trying to understand the present situation at IQSOFT and identifying the real needs followed. I also made a survey about ISO certification-process (in Hungary).

In Annex 3, a short overview of the evolution of building and introducing a software QMS at IQSOFT is given. Most of the work has been done by own initiative, without

---

3 Hungarian advisors are not very popular, mainly because of the general opinion that they don’t have enough experience in the field.

4 See some details in Annex 1.

5 Former = before 1989/1990; characterised mainly by isolation from world-wide trends.

6 It would be interesting to make a comparison between QMS-related activities suggested eg. in [Gillies92] pg. 161-163. and activities carried out at IQSOFT in the same purpose. This can be subject of further work.
being formally integrated in IQSOFT organisatoric structure. Discussions with the technical director and 4 other colleagues were of a real help.

I’d like to mention here the events/results that I consider to be most important.

a. Successes

<table>
<thead>
<tr>
<th>Nr.</th>
<th>Description</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Identifying and describing a general process life-cycle model by analysing different life-cycle models used at the company. The model has both system development and project management elements, offering a “menu” from which one can select elements considered to be important in the concrete case.</td>
<td>Elements being used from time to time by 1-3 teams. The model has been worked out using the MSProject tool. The general project life-cycle, the pre-defined resource pool etc. are helpful in project plans worked out with the same tool. Not enough feedback for improvement.</td>
</tr>
<tr>
<td>2.</td>
<td>Identifying and organising a directory structure for the projects. Mainly following previously mentioned model. Help files provided for every document needed. Help provided for the sequence of activities to be done.</td>
<td>Elements being used by 2-4 teams. There seem to be many unused elements. Continuously improving.</td>
</tr>
<tr>
<td>3.</td>
<td>Building document templates for the most important documents (12 available). Help facilities embedded.</td>
<td>Being used (partly or entirely) by approx. 8 employees. Feedback-based improvement.</td>
</tr>
<tr>
<td>4.</td>
<td>Building and filling in data in an internal information system (based on Doktár product), reflecting the structure of project steps and documents connected to each step as described above.</td>
<td>Being used. Offers a rapid way for the management to get up-to-date information about projects. By the beginning of 1995 approx. 15 projects having some or the entire information in the system. Some (technical) problems related to the maintenance of the database.</td>
</tr>
<tr>
<td>5.</td>
<td>Writing documentation for describing: -Configuration of hardware and software used at the company -Methods and (internal) rules regarding software and hardware maintenance -Customer and supplier list</td>
<td>Being used. Writing down helped to clarify current situation, suggestions came during description.</td>
</tr>
</tbody>
</table>

Table 1 (to be continued): Successes in developing and introducing software QMS at IQSOFT

---

7 At IQSOFT “teams” are formed mainly for carrying out a specific task, and are rather small (1-5 people).
a. Successes (continued)

<table>
<thead>
<tr>
<th>Nr.</th>
<th>Description</th>
<th>Remarks</th>
</tr>
</thead>
</table>

7. Software QM courses organised by IQSOFT for outsiders, based on own experience in introducing QMS.

Employees of the company were also interested in these courses. IQSOFT starts to being known for it's quality-oriented activities in the Hungarian market.

Table 1 (cont.): Successes in developing and introducing software QMS at IQSOFT

b. Failures

<table>
<thead>
<tr>
<th>Nr.</th>
<th>Description</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Quality management system handbook</td>
<td>Too general, concentrating too much on ISO prescriptions. Hard to understand and thus frightening to people not familiarised to ISO terminology.</td>
</tr>
<tr>
<td>2.</td>
<td>Description of methods and tools used at IQSOFT</td>
<td>Just a first-cut description exists. Mainly because of try to find and the difficulty of finding methods/methodologies and tools possible to be used by every team/for every customer.</td>
</tr>
<tr>
<td>3.</td>
<td>Description of internal organisation, functions, roles and responsibilities</td>
<td></td>
</tr>
</tbody>
</table>
| 4.  | Bootstrap-assessment carried out in April 1994. 2 projects and the overall organisation assessed. | One of the two assessed projects (A) followed prescriptions of the existing QMS, the other (B) didn’t. (A) has been qualified being 0.25 “better” than (B). Although reaching level 3 from 2 in the CMM is a huge step, this difference was not enough to show the usefulness (in ISO or CMM terms) of the QMS. Reason for (B) almost equalling (A) (in terms of CMM) can be found in the personality of (B) project manager, who is a “very good professional”

Table 2 (to be continued): Failures in developing and introducing software QMS at IQSOFT

---

8 This seems to correlate with results of research carried out in order to characterise “very good professionals”. See [Sonnen95].
b. Failures (continued)

| 5. | Itself the desire to introduce QMS overall the company. | This, in fact, cannot be characterised as an "absolute failure". It reflects the mentality of the management and the employees to accept only really helpful changes, despite the desire to get registered according to ISO. To be commented in the following. |

Table 2 (cont): Failures in developing and introducing software QMS at IQSOFT

1.3. THE THEORETICAL APPROACH

Although a theoretical background would have been needed from the beginning, because of situation mentioned in 1.1. theoretical approach followed the practical one. I have done the work mentioned in 1.2. parallel to a literature-research in software quality, in order to understand the changes that have been bought to this field by issuing the ISO Series.

I came to the idea that ISO strongly suggests a process-oriented approach (or manufacturing-oriented view\[9\]), while former approaches tried to focus on software-product characteristics (or product-based view\[10\]). In the question about the reason of this changing in the viewpoint I agree with the following statement: "[The manufacturing-oriented view] ... is attractive because it is the easiest view to quality." [Gillies92] I focused on the necessity of using structured system-development and project management methodologies, together with their computer-aided tools. The reason of focusing on them was my belief that using a structured, well documented way of working, having a well defined process life-cycle-model (which covers the entire life-cycle, not only the parts being present in a structured system-development and a project management methodology ) will lead to introducing the concepts of quality management, at least in ISO-terms.

To be noted that system development and project management methodologies and their tools are available in Hungary. Particularly, at IQSOFT Oracle*Case Method and SSADM\[11\] (system development), respectively Artemis Prestige and MSProject (project management) methodologies are available, together with their computer-aided tools. Majority of the employees are familiar with the concepts of these structured methodologies, and some elements are sometimes used (further details in [Balla94]).

In about half a year it became obvious, that a software QMS, but not even a structured system-development or project management methodology can be introduced at once. Applying every prescription would entirely change the company's activity, causing financial and moral crisis.

I understood in about one year that there were some elements in software quality management (and in ISO 9000 prescriptions as well) that can be introduced in a

---

9 I used Garvin's definition, [Garvin84]
10 Same as in previous footnote.
11 SSADM has been translated into Hungarian and is the most probable to become the recommended system development methodology in Hungary.
software company without major changes, thus they are more easily accepted by software developers. These elements were: configuration management (concept and plans), test plans and test journals, journals of project-related activities and documents; (mainly informal) quality reviews; concept and use of an informational system; existence and use of different document-templates with prescribed outline and contents. (For a more detailed description of these ideas see [Balla94].)

I noticed one basic difference between quality management -as done earlier - and quality management in ISO 9000-terms: *ISO 9000 certifies the quality system used by the supplier, not the outstanding quality of the product.*

I felt that the software product itself with it’s quality characteristics disappears somehow in the “jungle” of the process. I only tried to concentrate on some aspects which I considered to be elements in the process-oriented approach connected in a way to the product (eg. testing).

2. Some ideas about what and how could it have been done

Although it is very difficult to make statements about cause of failures or to state “what and how should/could it have been done” in a better way, I’ll try to formulate some suggestions regarding the mentioned aspects, using ideas, concepts, experience gathered - as it was possible in a period of 6 weeks - during my stay at the Technical University Eindhoven. These ideas are based on literature consulted and on discussions with Dutch software quality management specialists at TUE and at different software companies (see Annex 4.)

2.1. WHAT WAS WRONG IN THE INITIAL APPROACH

A. We tried to define “overall quality” for the company.
B. We identified this “overall quality” with an almost exclusive process-oriented approach.
B. Starting from the need of getting ISO 9000- certified in the shortest time possible, the concept of software QMS (in ISO 9000-terms) has not been adapted to the particular situation at IQSOFT.

Instead, it would have been better:
A. To concentrate on some (not too many) projects or products, to identify meaning of “quality” in each specific case, to identify some (not too many) quality-goals, concentrate on them - and then try to think about possibilities for assuring the specific goals. (At EXACT company, for instance, this goal is “integrity of data and databases provided to the users”.)
B. To recognise the division of the quality framework in objects and perspectives, and to understand the interactions between these aspects, in order to understand the general framework of quality factors in software production instead of concentrating only on processes.
C. Trying to tailor ISO 9000 prescriptions in concordance with the specificity of work done at IQSOFT instead of trying to build a “generally applicable” QMS, starting from ISO 9000-prescriptions.

Suggestions for solving problems sketched above (A., B., C.) can be found in [Trienek92], [Trienek&94], [Kusters&93].
In the following I’ll try to present some concepts and ideas used in the mentioned references, that in my opinion could be used at IQSOFT.

2.1.1. Useful concepts, with remarks

**Objects of software production:** “things” or “artefacts” in that some party that is involved in software production is interested. A quality factor has to be interpreted as a property or a set of properties of a software production object.

Objects can be classified in 3 distinct groups:

a. **Products:** include final deliverable products

b. **Processes:** any activities of persons involved in the production of software

c. **Resources:** any objects used by processes excluding products of other processes.

A quality factor has to be interpreted as a property or a set of properties of software production objects. In fact a software quality factor is an objectively or subjectively defined function over one or more “base” parameters, which are also called **quality attributes** or **quality criteria**. Examples of **quality factors** for the above mentioned object-groups:

a. Products: usability, reliability, maintainability

b. Processes: number of errors of a specific type, effort, cost

c. Resources: experience, skills like communicative ability etc.

<table>
<thead>
<tr>
<th>Remark 1:</th>
</tr>
</thead>
<tbody>
<tr>
<td>To be noted that at IQSOFT group b. has been emphasised, while group a. and c. have been almost totally ignored. Group c. has been present in a different approach, because of it’s connections with b</td>
</tr>
</tbody>
</table>

Each category of objects can be approached from three different **perspectives**. From each of these perspectives different approaches will be used to determine the **quality factors** of the different objects, and subsequently their “base” parameters and appropriate **metrics**. In Table 3. the different approaches of the distinct perspectives are shown.

<table>
<thead>
<tr>
<th>object</th>
<th>perspectives</th>
<th>Product</th>
<th>Process</th>
<th>Resource</th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td>user satisfaction based</td>
<td>participation based</td>
<td>interaction based</td>
<td></td>
</tr>
<tr>
<td>engineer</td>
<td>system based</td>
<td>design based**</td>
<td>application based</td>
<td></td>
</tr>
<tr>
<td>project manager*</td>
<td>project based*</td>
<td>control based*</td>
<td>allocation based*</td>
<td></td>
</tr>
</tbody>
</table>

* present at IQSOFT
** weakly present at IQSOFT

Table 3: The quality framework, objects and perspectives

[Trienek92], pg. 3.

<table>
<thead>
<tr>
<th>Remark 2:</th>
</tr>
</thead>
<tbody>
<tr>
<td>At IQSOFT the “object” was almost exclusively the “process”. The “perspective” was almost exclusive the “project manager” perspective.</td>
</tr>
</tbody>
</table>
The main categories of activities in quality-approach in software production are:

a. **Product modelling** category: concerned with modelling the concepts relevant to producing a software system or the “What” to be done.

b. **Process modelling** category: concerned with the modelling of the software production processes itself or the “How”, the “When” and the “By Whom” of the software production.

c. **Managing the reuse of previous work**

In order to differentiate between the specific characteristics of work done in the software company, the typology for customer orientation from industrial production control research is adapted to the software industry. Based on the characteristics of the activities in the primary production processes three types of software production were defined in Holland, TUE, respectively:

a. **Engineer-from-components**: product variety is restricted to a number of pre-selected families of products. Products or their components are market oriented developed instead of customer oriented developed.

b. **Engineer-from-products**: software production is focused on specialisation in specific application areas. Requirements of the client of the software product are restricted to a limited product range.

c. **Engineer-from-scratch**: there is no specialisation in typical products, purely production capacity is sold to the customer.

Characteristics of the different types of software production are presented below.

<table>
<thead>
<tr>
<th></th>
<th>Engineer from components</th>
<th>Engineer from products</th>
<th>Engineer from scratch</th>
</tr>
</thead>
<tbody>
<tr>
<td>Design and implementation specification</td>
<td>Analysis, design and implementation specification</td>
<td>Requirements, analysis, design and implementation specification</td>
<td></td>
</tr>
<tr>
<td>Formal product components</td>
<td>Informal product reference models</td>
<td>****</td>
<td></td>
</tr>
<tr>
<td>Formal activity structures **</td>
<td>Informal life cycle reference models</td>
<td>****</td>
<td></td>
</tr>
<tr>
<td>Activity modelling **</td>
<td>Life cycle modelling *</td>
<td>Life cycle modelling *</td>
<td></td>
</tr>
</tbody>
</table>

* Present at IQSOFT
** Partly present at IQSOFT

Remark 3:
At IQSOFT process modelling has been emphasised, product modelling and reuse of previous work were almost totally ignored, although there is a large amount of reuse, mainly in the field of object-oriented programming and Oracle-based applications.

Table 4: Characteristics of the different types of software production
[Trienek92], pg. 6.
After understanding the concepts and the possibility to apply them in the specific situation of a company, the **basic quality requirements** can be formulated and the **design of the quality management system** can follow.

For defining the quality requirements, attributes suggestions can be found in [Gillies92] and [Fenton92].

[Gillies92] presents a hierarchical model of quality. Quality factors have to be identified, each of them having one or more quality criteria. In order to implement quality, metrics are associated with each criterion. (See [Gillies92], chapter 2.)

[Fenton92] gives a table (pg. 44.), in which software entities are presented, in connection with their (internal and external) quality attributes.

Both approaches deal with too many attributes, the goal being the identification of the more possible quality attributes. In terms of “quality goal” such an amount of attributes can be less efficient, therefore I find very useful the approach suggested in [Kusters&93].

This approach uses the following main steps:

**a. Quality control loop** in the phase of a project: the main activities and their relationship are shown in Figure 1.

---

**Remark 4.:**

IQSOFT activity can be characterised as mainly engineering-from-products (mainly Oracle-based applications for specific customers, eg. government, banks, newspapers), some engineering-from-scratch (database requirement analysis and design, information systems implementation eg. for Labour Centre, Trigon Pharmaceuticals) and sometimes engineer-from-components (mainly in subjects connected to international research). For details see Annex I pg.2. However, this is only a first-cut-idea. I think, different types of software production at IQSOFT cannot be differentiated very clearly, and this differentiation could be done after further investigation.

To be noted that elements from Table 4. marked as “present at IQSOFT” (*) and “partly present at IQSOFT”: (**) are all connected to process modelling. Also, for the main type of activity (engineer-from-products) only elements connected to process modelling are present.
See description of the main activities in [Kusters&93]
To be noted that in this approach quality requirements will have to derive from the organisational context that the users in question operate in.
Relating quality requirements to quality factors requires a translation between the requirements set out by the users in their terminology, and the factors that will have to be in a language that developers are familiar with.

**Remark 5:**
At IQSOFT we did not concentrate on activities 2. and 4. Activities 1. and 3. have been weakly present. Our approach is more likely the one presented in [Kusters&93], as being the “present approach to quality”: a circular course, consisting of 3 steps (factors, measures, attributes), although identification of quality factors has been done in a general approach, following ISO-suggestions.

b. **Tools and techniques** suggested in determining quality factors:
- checklist, based on a questionnaire;
- a list with relevant prioritisation criteria;
- a relation matrix (in order to link requirements to factors).

c. **Matrix** depicting the relation between quality requirements and factors:
The goal of this matrix is the visualisation of the relationship between the quality requirements as vocalised by the users on the one hand and the quality factors needed by the developers on the other. The vertical axis of the matrix will contain the user requirements. The horizontal axis will contain the (technical) quality factors required by the developers. Details in [Kusters&93], pg. 8-9.

**Remark 6:**
At IQSOFT these 2 elements (tools and techniques /matrix) were not used. Checklists, list with relevant prioritisation criteria could be used. Matrix could be used after gaining some experience.

A method / methodology similar to this approach has been worked out in The Netherlands beginning with 1992, as a result of a project financed by the Government12. One of the basic ideas developed in this project was the idea of need to communication about quality (tools, methods, concepts), by involving the users. Questionnaires have been worked out to support the interviews. As results of interviews external attributes for software quality are identified (max. 3-5 attributes), followed by a translation into internal attributes. The developed methodology also gives a description of each attribute. A very important result of this work is the fact that user and developer agree about meaning of quality in a specific case.

---

12 According to a discussion with dr. ir. J.J.M. Trienekens. Reference material [Heemstra&94] is available only in Dutch.
For designing the QMS we have now the main concepts, we understood the specificity of the company's activity and we have understood - in common with the user - the most important quality attributes. Designing the QMS will have to be done according to this knowledge. According to [Trienek92], quality management differs according to the type of the activity done at the company.

<table>
<thead>
<tr>
<th>Type of activity</th>
<th>QM characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Engineer-from-scratch software production</td>
<td>Interaction-based approach to determine quality factors of human resources (e.g. skills and experience of developers to stimulate co-operation.) Strong capability orientation.</td>
</tr>
<tr>
<td>Engineer-from-products software production</td>
<td>Product quality management is primary. Main indications for quality management could be derived from both a user satisfaction and a system based approach of the determination of the quality factors.</td>
</tr>
<tr>
<td>Engineer-from-components software production</td>
<td>Quality management based on the determination of process quality factors. Quality management will have a strong process orientation.</td>
</tr>
</tbody>
</table>

Table 5: Characteristics of quality management for different types of software-producing activities [Trienek92], pg.7.

 ISO 9000-3 doesn't distinguish between different types of software producing organisations. By acknowledging different types of software producing organisations and based on their software production characteristics in [Trienek&94] different ways for addressing ISO 9000-3 elements are pointed out. For each ISO 9000-3 framework categories (quality system framework, life cycle activities, supporting activities) examples of interpretation and applications of standards are given.
3. Suggestions for different activity fields connected to software QMS

In this paragraph I'll concentrate on some activity fields that, according to the Bootstrap-assessment carried out at IQSOFT company (see Table 1. and 2.) seemed to be missing or done in a wrong way. These activities were: testing, risk management, applying of software metrics. I'll try to present some ideas as resulting from documentation and from discussions with Dutch specialists in different fields. In 3.4. and 3.5. some Case-tools are presented, which, in my opinion, could be adapted or used for the actual Hungarian situation.

3.1. TESTING

Need for a well done and efficient testing at IQSOFT emerged from 2 directions. 
- Bootstrap assessment pointed out lack of testing. The cause of this lack can be found mainly in users' behaviour, who prefer to have a working prototype in a shorter time and test it by themselves instead of having a professionally tested product in a longer time. (Of course, some testing activities exist at IQSOFT, carried out mainly by those who write the programs. These activities are ad-hoc and not at all or very little documented. The way and amount of testing highly depends on the project managers and the team-members. At IQSOFT there is no testing department.)
- Beginning with fall 1994 need for testing software products made by third parties emerged. Some Hungarian companies (not only software) would like their software products to be tested and qualified by "good professionals". (The fact that they contacted IQSOFT with this problem shows the company's reputation in the Hungarian software-market.) Although no testing/qualifying has been carried out yet, need for testing methods, techniques and tools can be noticed. The management also takes into consideration establishing a separate testing team.

In the following I'll present some concepts and ideas connected to testing, as resulting from discussions and demonstration at Exact company, Delft and from consulting [Zijder95], [Taguchi88],[Gintell&95], [Marzano&95], [MacDon&95], [Helping&95], [Clayton95]. At Exact company management is focusing on quality aspects. A separate quality management department exists, testing is done not by same peoples who wrote the programs. Lack of specific knowledge (testing, quality assurance and management etc.) is
surmounted by internal training: every employee is prepared for the very job he will have to
do during a 2-moth period. Employees have the possibility to follow training courses
regularly. The management also has mechanism to test employees' professional knowledge
from time to time.

Remark 10:
Exact started to function in 1984 as a small company (6 people), and managed
to be by 1994 among the strongest software companies in The Netherlands
(700 employees). Although it cannot be compared with IQSOFT in terms of
dimensions, similitudes can be found in the problems arisen.

A graduate student from Eindhoven University of Technology has the task to work out and
document testing methods and techniques, to insure use of testing tools, for the moment in
connection with the "Finance' banking system software. (The software has been developed
in programming language Progres.) Although the programs have been "black boxes" for the
tester, for the moment the company managed to have a well organized and documented
testing-procedure.

[Zijder95] states that basic element in testing is to understand **what should be tested?**
Other elements in making a good test-plan; understanding what the customer wants; paying
attention to data consistency; understanding the program-flow; graphical presentation of
the user interface; understanding of the way in which the system operates (communication
with other systems, devices, printers, screens etc.)
The "understanding" mentioned above is based on knowledge that exists basically in
people's mind, but can be modeled using different techniques. The "written knowledge" can
be used in forthcoming projects.

Remark 11:
If a software is being developed using a structured system development
methodology, many elements of the above mentioned "knowledge" will be
formalized (written) before the testing process begins. In such case the tester
would have an easier job. Otherwise he/she has to do some reverse
engineering, for building diagrams, charts etc. starting from a program-code.

A good test plan should be done only after understanding the above mentioned aspects. It
has to cover all the important elements in the system, and has to be able to ensure that
tested elements, test procedures will be documented.
Decision regarding the testing techniques, procedures and tools to be used can be taken
only after the above mentioned problems have been understood.
Some testing techniques: source-code analysis, usage analysis, use of orthogonal arrays,
flowcharts and other diagramming methods, program slicing, system capability test.
Testing tools used at Exact company:
**HighTest** (automatisation of testing the applications running under MsWindows, with log-
facility)
**Wintrace** - developed at Exact; keeps track of activated programs.
**Ortran** - developed by a Russian company for Exact; generates test-data, based on theory
of orthogonal arrays. (Further information about orthogonal arrays in [Taguchi88])
3.2. RISK MANAGEMENT

According to the results of Bootstrap-assessment carried out at IQSOFT in April 1994, use of risk management concepts and practice is missing at the company. I’ve spent a lot of time consulting available documentation, and came to the idea that in order to have risk management we should:

1. Find an adequate (mathematical) model, identify meaning of the model’s each parameter for IQSOFT situation and build up the model, which will help us to predict risk.

2. Record the results of projects carried out with using the model described. The more experience (projects) we will have, the more precisely risk will be predicted. This approach gave no results, basically because of the difficulties in identifying an “adequate” model and in understanding the meaning of it’s parameters for IQSOFT.

[Kusters93] suggests an other approach, based on the experience gained in a study which took place in a large Dutch governmental organisation. The main concepts of this approach are:

- Risk analysis: identifying the potential risk in each case.
- Risk monitoring: identifying activities with which the risk could be diminished.
- Risk ex post evaluation: at the end of the project; storing connected data in order to be used in forthcoming projects.

For identifying the potential risk factors [Kusters93] suggests the use of checklists, with not more than 30-40 questions. (The idea of checklists is also promoted by SEI - see [Carr&93], but this approach suggests to concentrate on all possible risk elements. The SEI-checklist has more than 300 questions.)

The checklist proposed by [Kusters93] concentrated on as few risk elements as possible. It consisted of 9 clusters of risk factors (principal, user, user management,
type of application, specification, project management, development staff, means-tools-hardware, implementation and control). In each cluster essential risk factors have been identified. The checklists have been completed during interviews carried out in each case with approximately 10 people. During the study it became obvious that not only risk managers have to be asked about risk, members of the project team and users also have to be involved.

Identifying risk factors is recommendable to be done by independent specialists (risk analyser).

After identifying risk factors, risk analysing and risk monitoring meetings have been carried out, in order to analyse efficiency of risk monitoring and risk management activities done in the project. Existence of earlier identified risk factors and their priority also has been analysed. Users have been involved in these meetings.

**Remark 13:**
Use of the suggested approach (checklists, interviews, meetings) could be benefice at IQSOFT. Group aspect, communication commitment (within the team and with the user), “keep it simple/keep it small” ideas will probably give better results than the formerly tried way (identifying a mathematical model for predicting risk). The more projects will be carried out by using risk identification/prediction activities, the more efficient risk management will be. Connected data have to be kept in order to use in forthcoming projects.

### 3.3. USE OF SOFTWARE METRICS

According to the results of Bootstrap-assessment carried out at IQSOFT in April 1994, use of software metrics is missing at the company. The reason can be found in lack of definition of specific quality factors/quality goals. Concentrating on every possible quality factor/attribute will lead to failing in defining quality goals. The “overall quality” cannot be a goal.

According to [Fenton92] and [Gillies92] software metrics should be associated to specific quality attributes. In [Fenton92] many suggestions can be found about using software metrics.

According to [Pfleeger95], use of metrics can be useful but can also lead to no result. It is very important to know what we measure, why we measure and what are we going to do with the results. Some metrics are not relevant in terms of suggesting ways for improvement. (Eg.: SEI CMM quantifies an organisation being on the level 1.8, that means “almost 2”. The difference of 0.2 gives no guideline for reaching a higher level.) Complexity of applying different metrics also has to be taken into account. If, for instance, a program is only characterised in terms of “lines of code”, other aspects (e.g. number of loops) are being neglected, which will lead to misjudging the quality.
In [Gillies92] we can find some statements about “What makes a good metric?”

<table>
<thead>
<tr>
<th>Objectivity</th>
<th>The results should be free from subjective influence.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reliability</td>
<td>The results should be precise and reliable.</td>
</tr>
<tr>
<td>Validity</td>
<td>The metric must measure the correct characteristics.</td>
</tr>
<tr>
<td>Standardisation</td>
<td>The metric must be unambiguous and allow for comparison.</td>
</tr>
<tr>
<td>Comparability</td>
<td>The metric must be comparable with other measures of the same criterion.</td>
</tr>
<tr>
<td>Economy</td>
<td>The simpler and, therefore, the cheaper the measure is to use, the better.</td>
</tr>
<tr>
<td>Usefulness</td>
<td>The measure must address a need, not simply measure a property for its own sake.</td>
</tr>
</tbody>
</table>

Table 6: [Gillies92], pg. 39.
Seven criteria for a good metric, after Watts, 1987

3.4. CONFIGURATION MANAGEMENT TOOLS

Configuration management is a concept which seemed to be useful and thus applicable at IQSOFT. We tried to identify project configuration elements, to record them and to record information about them, to define milestones in projects in which the entire project configuration has to be archived. Details about configuration management plans at IQSOFT can be found in [Balla94].

During my stay at the Technical University Eindhoven I attended a presentation of a configuration management tool, at Cap Volmac company, Utrecht.

The tool (Perform) is now in the testing phase. It has been worked out according to ITIL\textsuperscript{13} standards, within the ITIM\textsuperscript{14} project. The tool can be used with different platforms, eg. Windows.

Perform is in fact a 5-module support set. The main features of the modules are:

- **Configuration management**: administration of the composition and the actual status of the IT facilities. All elementary parts of the IT facilities are identified and recorded as configuration items.

---

\textsuperscript{13} ITIL = Information Technology Infrastructure Library

\textsuperscript{14} ITIM = Information Technology Infrastructure Management
Help desk / Problem management: administration of incidents and problems, and prevention of problems.

Change management: administration and management of change requests on the IT facilities.

Software control and distribution: control of environments and libraries and distribution of (versions of) software modules and documentation through the environments.

3.5. COST ESTIMATION TOOLS

During my stay at the Technical University Eindhoven I attended a presentation of a cost estimation tool\textsuperscript{15}, at Cap Volmac company, Utrecht. Cost estimation is done at IQSOFT, but mostly based on experience, analogy. Some estimation is made using MSProject tool, but this is only a cost estimation concentrating on resource usage and afferent costs.

ESTEEM is a tool for “reliable cost estimates through experience”. It’s working principle is based on the statistical method “linear regression”. After defining a configuration in which cost estimation is to be done (life cycle, milestones etc.) it will be able to make cost predictions in a 95% confidence interval. The estimation is done on an “attribute”\textsuperscript{16} level. It gives information about the most important cost factors. It uses data of previously made estimations, having the possibility to find common factors in projects. Estimation data is kept in a repository. The more estimations are maid, the more confident prediction will be\textsuperscript{17} (in case of projects having similar elements).

It is shown that ESTEEM-data has to be specific for each customer. Using data from other companies will not give the desired results.

Remark 16:
ESTEEM could be used at IQSOFT and in other Hungarian companies. It offers a different approach to cost estimation because of the linear regression model used. It could also be adapted for use in companies different from software companies (further investigation is needed to understand the specificity of the life cycle that can be modelled). The tool will be presented at TUB and IQSOFT in April 1995, within the Tempus-CUSE project.

\textsuperscript{15} The tools has been presented in a framework of a system development and project management toolset (SDW). I concentrate on ESTEEM only because tools similar to SDW are available in Hungary. IQSOFT is distributor in Hungary of Artemis Prestige project management tool.

\textsuperscript{16} Attributes are defined by the user. Can be: overall, milestone, steps etc.

\textsuperscript{17} It requires at least 7-8 project-data in the repository for a confident estimation.
4. Conclusions

Software quality management concepts and ideas have to be used according to the specific situation of every software company. Business situations, characteristics of software developed at a company has to be taken into account. ISO 9000-3 prescriptions have to be tailored to fit the company’s needs. Software quality oriented work has to begin with understanding of concepts. Thinking about methods, techniques and tools can follow these step. Application of methods, techniques and tools does not solve the real problems connected to software quality. Organisations should be aware of the fact that activities of quality managers, testers, tools - and - methodology consultants require persons working full time in the area.
IQSOFT Profile (January 1995)

IQSOFT
Medium-size software development and trading company in Hungary

Full name
IQSOFT Intelligent Software Computing Technology Development
Manufacturing and Trading Co. Ltd.

Shareholders
IQ Management Ltd. (employees) 12.6 MHUF 54%
Bank Austria AG 6.0 MHUF 25%
West LB Hungary 2.4 MHUF 10%
Private (employees and others) 2.8 MHUF 11%

History
1977 Department of state owned Computer Research Centre (SZKI)
1990 Spin-off company created, majority owned by SZKI
1993 Management/employee buy-out of IQSOFT

Basic figures
<table>
<thead>
<tr>
<th>year</th>
<th>staff</th>
<th>revenues</th>
<th>profit</th>
<th>dividend</th>
</tr>
</thead>
<tbody>
<tr>
<td>1990</td>
<td>35</td>
<td>77 MHUF</td>
<td>8 MHUF</td>
<td>15%</td>
</tr>
<tr>
<td>1991</td>
<td>43</td>
<td>160 MHUF</td>
<td>13 MHUF</td>
<td>25%</td>
</tr>
<tr>
<td>1992</td>
<td>56</td>
<td>246 MHUF</td>
<td>13 MHUF</td>
<td>25%</td>
</tr>
<tr>
<td>1993</td>
<td>58</td>
<td>360 MHUF</td>
<td>4 MHUF</td>
<td>20%</td>
</tr>
</tbody>
</table>

Staff
- 20-29 years: 16, University: 49, Management: 4
- 30-39 years: 14, Secondary: 9, Technical: 37
- 40-49 years: 21, TOTAL: 58, Commercial: 10
- over 50: 7, Men: 36, Administrative: 7
- Women: 22

TOTAL 58

Infrastructure
building 700 sqm rented office space (rooms for 2-4 persons)

hardware
- Sequent 2000/250 server (2-with six processors altogether)
- SUN workstations (3)
- 486/386 based personal workstations (50)
- 386 based PCs in employees homes (30)
- HP printers, MO, CD, scanners

software
- UNIXs, Novell, OS/2, Windows Workgroup,
- Ms Mail, Ms Office, WinProject
- ORACLE, GUPTA, MProlog, C and C++

network
- Ethernet LAN, X-25 connection
- Compuserve access
- Internet

Address
H 1142 Budapest Teleki Blanka-u 15-17
Phone: (361)-251-5449, (361)-251-5949
Fax: (361)-163-7416
E-mail: iqsoft@iqsoft.hu

Annex 1 (pg.1.):
Short information about IQSOFT Ltd.
IQSOFT Activities (January 1995)

Software development

Bank Austria

Expert system shell (in MProlog)
Broker information system
Banking data processing, etc.

for export

Software distribution in Hungary

1989-1993:

• Sole distributor of Oracle products in Hungary
• Dealer, VAR and "system house" co-operation with Oracle Hungary

Since mid 1993:

• Distribution of database applications and tools
  - Oracle*Libraries (UK)
  - Avalon CII M (USA)
  - Ora*Care (CND)
  - System Access (Singapore)
  - RZW accounting package (A)
  - Personnel and Pay-roll System (H)
  - Gupta: SQLWindows (USA)
  - Sequent (USA)

Services:

- Localisation
- Adaptation
- Support
- Training
- Consultancy

Document imaging and office automation

Distribution of Windows based products developed by Hypermedia Systems (H):

- document imaging and archiving package DOKTÁR,
- law databases on CD-ROM

System integration with DOKTÁR and Microsoft products

Hungarian application development projects

Database requirement analysis and design

Information systems implementation

Production control system development etc.

International research

Parallel logic programming
Knowledge based systems

Networking survey and analysis
Quality management

Annex 1 (pg.2.):

Short information about IQSOFT Ltd.
<table>
<thead>
<tr>
<th>Year</th>
<th>Theme</th>
<th>Researchers</th>
</tr>
</thead>
<tbody>
<tr>
<td>1973</td>
<td>test-data generators</td>
<td>Várkonyi, Zsolt</td>
</tr>
<tr>
<td></td>
<td>test bed</td>
<td>Soós, Klára*</td>
</tr>
<tr>
<td>1975</td>
<td>Qualisoft - system</td>
<td>Szentes, János</td>
</tr>
<tr>
<td>1976</td>
<td>Formal description of software components by Structured Abstract Models (+VDM)</td>
<td>Dömölki, Bálint*</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sántáné, Tóth, Edit*</td>
</tr>
<tr>
<td>1976</td>
<td>Verification of programs</td>
<td>Várkonyi, Zsolt</td>
</tr>
<tr>
<td>1977</td>
<td>Basic Quality Requirements for Software</td>
<td>SZKI **</td>
</tr>
<tr>
<td>1978</td>
<td>Qualifying user-programs according to their source code listing</td>
<td>Beiczer, Odón</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Szentes, János</td>
</tr>
<tr>
<td>1978</td>
<td>Applying correctness-measurer transformations for program synthesis and program optimisation</td>
<td>Farkas, Zsuzsa*</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sántáné, Tóth, Edit*</td>
</tr>
<tr>
<td>1979</td>
<td>Software metrics</td>
<td>Varga, László</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Kozma, László</td>
</tr>
<tr>
<td>1984</td>
<td>Methodology for pre-validation of programs</td>
<td>Bánné</td>
</tr>
<tr>
<td>1985</td>
<td>KAF - program sets for official institutes, qualifying programs</td>
<td>Kepler, Károly</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Gáspás, Mátéyás</td>
</tr>
</tbody>
</table>

* working now at IQSOFT
** SZKI = National Computer Science Research Centre; large projects; people working now at IQSOFT were involved in the projects

Annex 2:
Software quality - oriented - research in Hungary, 1973-1985\(^{18}\)

\(^{18}\) References given in [Balla94]
<table>
<thead>
<tr>
<th>Datum</th>
<th>Activity / event</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oct. 1993.</td>
<td>QMS documentation-set developed according to ISO prescriptions</td>
</tr>
<tr>
<td>4. Nov. 1993.</td>
<td>QM, QMS and ISO-certification concepts presented to all employees in a regular 3-months meeting. IQSOFT management presents the idea as being very important to the welfare of the company.</td>
</tr>
<tr>
<td>31. Nov. 1993.</td>
<td>IQSOFT management contacts Bootstrap organisation (assessment is offered cheaper until end of November), specifying that would like to be assessed only in April 1994.</td>
</tr>
<tr>
<td>7. Dec. 1993.</td>
<td>3 of the 10 above mentioned employees give their remarks on the QMS Handbook. General opinion: handbook is too general, concepts used are not well known, we should concentrate on the existing situation at IQSOFT. Question: what should the QMS cover: some specific projects, project-related-activities or all types of work done at IQSOFT? Solution: QMS should cover all project-related activities.</td>
</tr>
<tr>
<td>7. Dec. 1993.</td>
<td>In a discussion with the technical director requirement for an internal information system is formulated. The technical director suggests use of Doktar-product for identifying the information system. I begin to study the product's performances. Employees seem to be sceptic in use of Doktar. (The company sells it and offers assistance, but it has never been used in internal work.)</td>
</tr>
<tr>
<td>by 11. Jan. 1994.</td>
<td>The other 7 employees give their remarks about QMS Handbook. General opinion: too many &quot;papers&quot; needed. Question: in QMS documentation project management steps should also be present with their documents, or just system-development-steps and connected documentation? Solution: both activity types should be present in the QMS.</td>
</tr>
<tr>
<td>11. January 1994</td>
<td>According to a discussion with the technical director, actual QMS is going to be introduced in 2 projects (smaller projects, for databases) Suggestion for using SSADM at IQSOFT. Discussion about buying SSADM Engineer. In my opinion employees should be familiar with SSADM-concepts first</td>
</tr>
<tr>
<td>12. January 1994.</td>
<td>Discussion with (potential) project managers, supposed to use QMS. colleagues expected me to be a &quot;QM-specialist&quot; and to tell exactly what is to be done.</td>
</tr>
<tr>
<td>17. January 1994.</td>
<td>Further discussions with (potential) project managers, supposed to use QMS. Agreement regarding a recommended directory structure for projects (I'm going to work it out). (Theoretical) agreement about the usefulness of a project life-cycle-model, to cover all possible system development and project management steps. I shall work it out using MSProject.</td>
</tr>
</tbody>
</table>

Annex 3. pg.1.: Short overview of QMS-related activities at IQSOFT
18. January 1994. Recommended directory structure for keeping all project-related data available on the network. I used sequence numbers in order to provide help regarding the sequence of activities to be done. Every subdirectory contains a help subdirectory, in which help for outline and contents of recommended documents (in recommended sequence) are given.

24. January 1994. First project using QMS starts. Project plan made "with QMS" and "without QMS", using MSProject. It came out that using actual QMS-prescriptions will lengthen the project life cycle by approx. 1 month. (The project was supposed to finish in 2 months, without using QMS-prescriptions.) Activities like "project initiation", "reviews" etc. are planned to last 1-5 days.

27. January 1994. Discussions with 5 employees about documentation set prescribed by QMS. In the opinion of one employee using QMS lengthens project life cycle, but we will get back this time by having precise prescriptions for the whole menu and sequence of activities to be done.


31. January 1994. I finished working out the general project life-cycle model. I built a model using MSProject. The model - beneath activities and documents - has a predefined resource-pool, cost estimation. Time schedules have been made to indicate the percentage of each activity in the whole life-cycle.

1. February 1994. Discussion with the technical director. Problems: activities like writing tenders, offers should also be considered belonging to projects; some projects can happen to finish at this point; nevertheless, data gathered should be used in analysing causes of successes and failures in tenders. Agreement regarding the mechanism of "reporting" during a project. Agreement about archiving project configuration. Talk about archiving project documents using Doktar system.


22. February 1994. Project initiation phase review for the first project using QMS prescriptions. Review procedure and documentation done according to QMS prescription (Quality reviews at IQSOFT). One of the major risks for the project is the continuously changing user-specification.

8. March 1994. Discussion and agreement about structure of archiving system for projects. The structure will be built up by a Doktar-specialist.

12. March 1994. Doktar structure available on the network. I introduce the "dummy project" QMS, to provide help regarding the structure.


24. March 1994. Getting information about Hungarian translation of SSADM documentation set. This methodology is discussed (by governmental organisations) to become the recommended system development methodology in Hungary.

24. March 1994. On the 21. March meeting it became obvious that many colleagues are not familiar with QMS-oriented work done. I wrote a short description about actual state of QMS, and made it available in a shared internal mail folder.

Annex 3, pg.2.:

Short overview of QMS related activities at IQSOFT
<table>
<thead>
<tr>
<th>Date</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>6. April 1994</td>
<td>ProDoc documentation set arrives. I use some ideas (eg. embedded help facility) in constructing internal document templates, which correspond to internal QMS and ISO 9003 prescriptions.</td>
</tr>
<tr>
<td>11. April 1994</td>
<td>Attempt to join an international software QMS-research project (Copernicus, together with TU Maribor, TU Glasgow and BA). I worked out a project proposal, regarding the built of a methodology to help introducing ISO 9000 prescriptions for software developing organisation. (The project proposal has been accepted by the co-ordinator - British Airforce -, but the deadline for submitting the overall proposal has been missed.)</td>
</tr>
<tr>
<td>18. April 1994</td>
<td>The technical director informs by e-mail all employees about Doktar-QMS facilities, highly recommending their use.</td>
</tr>
<tr>
<td>27.-29. April 1994</td>
<td>Bootstrap assessment. 2 projects - formerly mentioned one, working with QMS prescriptions and an other one, working without QMS prescriptions - and the overall company assessed. Management and employees taking part in the procedure considered it being useful.</td>
</tr>
<tr>
<td>9. May 1994</td>
<td>Need for a Feasibility study, according to SSADM. I made the template, fitting internal standards.</td>
</tr>
<tr>
<td>25. June 1994</td>
<td>Revision of Doktar archiving structure, according to feedback.</td>
</tr>
<tr>
<td>June 1994</td>
<td>The first project using QMS prescriptions ends. Not all prescribed steps have been followed, mainly because of the customers wish to have a working prototype the sooner possible. The initial system specification had to be changed several times, in order to keep the customer. Change management procedures could not always have been followed due to time pressure. Nevertheless, the project manager found the QMS prescriptions to be useful, and agreed about the need of a structured, well organised way of working.</td>
</tr>
<tr>
<td>28. June 1994</td>
<td>Doktar-system (technical) problems. According to the survey made by this occasion, 15 projects (+QMS project) have more or less of their project-related data in the system.</td>
</tr>
<tr>
<td>29. June 1994</td>
<td>New potential teams, wishing to use QMS. (Oracle Libraries)</td>
</tr>
<tr>
<td>6. July 1994</td>
<td>Discussion with the technical director regarding actual state of QMS-project (building up and introducing QMS at IQSOFT). The initial deadline for getting ISO-certified (December 1994) cannot be reached. The cause: lack of suitable projects for introducing QMS. The management does not want to impose too strictly the use of the prescriptions. In the technical director's opinion there are many useful elements in the actual QMS.</td>
</tr>
<tr>
<td>8. July 1994</td>
<td>Final discussion regarding Bootstrap-assessment, with participation of the managing director. Testing, risk management, use of software metrics have to be emphasised. Discussing the possibility of setting up testing team(s) and a quality management department.</td>
</tr>
<tr>
<td>30 August 1994</td>
<td>Software QM lecture at the Technical University Budapest, for foreigner students participating in the Summer University courses.</td>
</tr>
<tr>
<td>September 1994</td>
<td>IQSOFT management wishes to hire a specialist to work out the IQSOFT Structure and Organisational Frame Handbook. The trial fails because of the high price asked. Decision: the handbook should be worked out by the quality manager, with the help of IQSOFT staff.</td>
</tr>
</tbody>
</table>

Annex 3., pg.3:
Short overview of QMS-related activities at IQSOFT
<table>
<thead>
<tr>
<th>Date</th>
<th>Event Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>October 1994.</td>
<td>Need for testing, formulated by other companies.</td>
</tr>
<tr>
<td>7.-9. November 1994.</td>
<td>Due to advertisement made at the Annual National Quality Week, more companies contacted IQSOFT asking information about the software-quality-oriented work done.</td>
</tr>
<tr>
<td>October-December 1994.</td>
<td>Finishing handbooks describing: hardware and software maintenance system at IQSOFT. Done together with the technical director and the system engineers.</td>
</tr>
<tr>
<td>December 1994.</td>
<td>IQSOFT (and TUB) takes part in the international Tempus CUSE project, connected to software engineering and having QM aspects. (Other partners: University of Karlsruhe, University of Brighton, Technical University Eindhoven.)</td>
</tr>
<tr>
<td>December 1994.</td>
<td>The Hungarian National Standard Institutes contacts IQSOFT quality manager in connection with a national project, which has the scope of identifying the quality-oriented knowledge needed and working out the handbooks which are to be used in the educating system. I submitted a proposal in the field of software quality management. The decision regarding acceptance of proposals will take place in March 1995.</td>
</tr>
<tr>
<td>January 1995.</td>
<td>In an interview given to the Hungarian issue of Computerworld, Balint Domolki, managing director of IQSOFT mentions getting ISO 9000 registration for IQSOFT as being one of the important goals for the company in 1995.</td>
</tr>
</tbody>
</table>

**Quality management system handbooks:**

<table>
<thead>
<tr>
<th>Title</th>
<th>Last Issue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standards, techniques and tools used at IQSOFT</td>
<td>20. Oct. 1993</td>
</tr>
<tr>
<td>Internal and external quality certification at IQSOFT</td>
<td>12. Nov. 1993</td>
</tr>
<tr>
<td>IQSOFT Software Documentation System</td>
<td>January 1995</td>
</tr>
<tr>
<td>Quality reviews at IQSOFT</td>
<td>15. Nov. 1993</td>
</tr>
<tr>
<td>Introducing QMS at IQSOFT</td>
<td>15. Nov. 1993</td>
</tr>
<tr>
<td>IQSOFT-QMS Glossary of terms</td>
<td>2. Apr. 1994</td>
</tr>
</tbody>
</table>

**Related handbooks, worked out within the same project:**

<table>
<thead>
<tr>
<th>Title</th>
<th>Last Issue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Roles, responsibilities at IQSOFT</td>
<td>15. Dec 1993</td>
</tr>
<tr>
<td>Hardware and software maintenance at IQSOFT</td>
<td>15. Jan. 1995</td>
</tr>
<tr>
<td>Description of IQSOFT internal information system</td>
<td>26. Apr. 1994</td>
</tr>
<tr>
<td>IQSOFT customer and supplier list</td>
<td>15. Dec. 1994</td>
</tr>
</tbody>
</table>

Annex 3., pg.4.
Short overview of QMS-related activities at IQSOFT
<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monday, January 16</td>
<td>14.00 - 16.00:</td>
<td>Jos Trienekens, Derkjan Lok, John Stienen, Stefan Uytegt</td>
</tr>
<tr>
<td>Thursday, January 19</td>
<td>13.30 - 17.30:</td>
<td>Workshop “Human Factors in Software Quality” (van der Schaaf/ Trienekens)</td>
</tr>
<tr>
<td>Friday, January 20</td>
<td>15.30 - 16.30:</td>
<td>Jos Trienekens (discussion about software qm systems, applying ISO 9000, articles to be studied)</td>
</tr>
<tr>
<td>Monday, January 23</td>
<td>10.00 - 11.00:</td>
<td>Thys Zijderveld (testing, methods and tools)</td>
</tr>
<tr>
<td></td>
<td>11.00 - 12.00:</td>
<td>weekly meeting (Trienekens, Lok, Stienen, Uytegt)</td>
</tr>
<tr>
<td>Wednesday, January 25</td>
<td>8.30 - 15.00:</td>
<td>Trienekens: visit to Brussels, discussions with Tom Flynn (Dublin) and E. Trodd (Brameur, London) about software quality management- connected projects, possibilities of Hungarian collaboration</td>
</tr>
<tr>
<td>Friday, January 27</td>
<td>11.00 - 13.00:</td>
<td>Visit to EXACT, Delft; discussions about testing methods and tools used + demonstration; discussion with 2 quality managers</td>
</tr>
<tr>
<td>Wednesday, February 1</td>
<td>10.00 - 11.00:</td>
<td>Interview given to a TUE student about Hungarian informational infrastructure</td>
</tr>
<tr>
<td>Thursday, February 2</td>
<td>9.00 - 10.00:</td>
<td>weekly meeting (Trienekens, Lok, Stienen, Uytegt)</td>
</tr>
<tr>
<td>Monday, February 6</td>
<td>13.00 - 14.00:</td>
<td>Jos Trienekens (discussion about Dutch approach to understand and apply software quality attributes, possibilities for Dutch - Hungarian collaboration etc.)</td>
</tr>
<tr>
<td>Tuesday, February 7</td>
<td>14.00 - 17.00:</td>
<td>Collocvium on software metrics</td>
</tr>
<tr>
<td>Wednesday, February 8</td>
<td>9.00 - 9.30:</td>
<td>Meeting Prof. Theo Bemelmans</td>
</tr>
<tr>
<td></td>
<td>10.00 - 11.00:</td>
<td>weekly meeting (Trienekens, Lok, Stienen, Uytegt)</td>
</tr>
<tr>
<td>Monday, February 13</td>
<td>10.30 - 11.30:</td>
<td>weekly meeting (Trienekens, Lok, Stienen, Uytegt)</td>
</tr>
<tr>
<td></td>
<td>14.00 - 17.00:</td>
<td>Cap Volmac (Utrecht) : configuration management tools (discussions, demonstration)</td>
</tr>
<tr>
<td>Tuesday, February 14</td>
<td>10.00 - 11.00:</td>
<td>Discussion about risk management (dr. Rob Kusters, TUE)</td>
</tr>
<tr>
<td>Thursday, February 16</td>
<td>13.30 - 16.30:</td>
<td>Cap Volmac (Utrecht): cost estimation tools (ESTEEM) - discussions, demonstration</td>
</tr>
<tr>
<td>Tuesday, February 21</td>
<td>10.30 - 11.30:</td>
<td>weekly meeting (Trienekens, Lok, Stienen, Uytegt)</td>
</tr>
</tbody>
</table>

Annex 4:
Meetings, discussions during my visit at Eindhoven Technical University
References:


1995.


* Consulted before visit to TUE
The following EUT-Reports can be obtained by writing to:
The costs are HFL 5.00 per delivery plus HFL 15.00 per EUT-Report (unless indicated otherwise), to be prepaid by a Eurocheque, or a giro-payment-card, or a transfer to bank account number 52.82.11.781 of Eindhoven University of Technology with reference to "Bibl.Bdk", or in cash at the counter in the Faculty Library.

20 LATEST EUT-REPORTS

EUT/BDK/67 Thematiek en methodologie in de organisatiekunde : een inhoudelijke verkenning over de periode 1986-1991 op basis van onderzoek van enkele Nederlandse tijdschriften
J.D. van der Bij, J.A. Keizer

EUT/BDK/66 Naar een tweede generatie total quality management
J.D. van der Bij, J.E. van Aken

EUT/BDK/65 Economische aspecten van informatietechnologie : de stand van zaken en de praktische relevantie R.M.H. Deitz

EUT/BDK/64 The Socio-Technical Systems Design (STSD) Paradigm : a full bibliography of 3082 English-language literature references
F.M. van Eijnatten, S.J.C. Eggermont, G.T.A. de Goffau, I. Mankoe

EUT/BDK/63 Het Socio-Technisch Ontwerp Paradigma van Organisaties : een bibliografie van 1145 Nederlandstalige literatuurreferenties
F.M. van Eijnatten, S.J.C. Eggermont, G.T.A. de Goffau, I. Mankoe

EUT/BDK/62 De service-mix : uitgangspunt voor succesvol relatiemanagement
H.W.C. van der Hart, M.A.M. Wollaert, J.P.M. Wouters

EUT/BDK/61 Ondersteuning van professionals m.b.v. IT M.W. 't Hart

EUT/BDK/60 Organisatievorm of basis van Groepentechnologie H.H. van Mal

EUT/BDK/59 The Socio-Technical Systems Design (STSD) Paradigm : A Full Bibliography of 2685 English-Language Literature References
F.M. van Eijnatten

EUT/BDK/58 Verbalization rate as an index of cognitive load
J.A. Brinkman

EUT/BDK/57 Trends and tasks in control rooms T.W. van der Schaaf

EUT/BDK/56 The system of manufacturing: A prospective study
J.C. Wortmann, J. Browne, P.J. Sackett

EUT/BDK/55 Rekenmodellen voor de grootschalige mestverwerking; gebaseerd op het MEMON-mestverwerkingsproces Mat L.M. Stoop

EUT/BDK/54 Computer, manager, organisatie (deel I en II)
R. Cullen, H. Grünwald, J.C. Wortmann

EUT/BDK/53 Risico diagnose methode voor produktinnovatieprojecten; Een uitwerking toegesneden op de Industriegroep TV van Philips Glas te Eindhoven/Aken J.I.M. Halman, J.A. Keizer

EUT/BDK/52 Methodological problems when determining verbal protocol accuracy empirically J.A. Brinkman

EUT/BDK/51 Verbal protocol accuracy in fault diagnosis J.A. Brinkman

EUT/BDK/50 Techniek en marketing H.W.C. van der Hart

EUT/BDK/49 Een methoden voor kosten-batenanalyse voor automatiseringsprojecten bij de overheid M. van Genuchten, P. Heemstra, R. Kusters

EUT/BDK/48 Innoveren in technologie-gedreven ondernemingen, bedrijfskundige aspecten van de voortontwikkelingsfunctie W.H. Boersma